Question4

Solution :

First, sort all the jobs by profit in decreasing named des_jobs and create a list that each index means a slot for working time named subjob_list.

Then, iterate on des_jobs. For each job, if there is an empty slot before the deadline in subjob_list, select the slot which is closest to the deadline but before deadline(the same as the largest index but small than deadline) an put it in. If there is no empty slot before the deadline, skip this job.

After iterating on the whole des_jobs, the subset of jobs that maximizes profit is subjob_list.

Proof :

If the subjob_list isn't the maximizes profit, it means that we have skipped a job A by error because there is no slot for it. So, in the right schedule, we should replace a job B by job A. But the position of job A and B is the same and it will not influence the next jobs, the profit of job A is no more than job B, the total profit will not increase. Thus, this solution is optimal.

The time complexity of sort is $O(nlog(n))$. Because we have n jobs and we need to check at most n slot so the second step is $O(n^2)$.

The total time complexity is $O\big(nlog(n)\big) + O(n^2) = O(n^2)$.